(7) *Text*: Text may be added to the picture. This text may be a simple string or it could be the data from one or more fields of the tuple.

6.3 Using INGRES Data

Each icon in the GDS may reflect the data in the entity it represents. This is done by making the data in the entity accessible from within the icon-class description. It is also possible to retrieve other tuples from any relation in INGRES, to aid in the description of an icon class. There are two statements for performing retrievals. One is the *get* statement. The get statement is used when one tuple is needed in response to a query, as in finding the beam of a particular ship. If more than one tuple satisfies the qualification of the statement, an error occurs. The other statement is a *for loop* statement which is used to retrieve a set of tuples from a relation. This statement uses one or more statements that are executed once for each tuple retrieved. The for loop construct could be used to retrieve the previous positions of a ship from a track history relation. Each position could then be displayed as part of the wake of the ship.

6.4 Subicons

Another feature of an icon class is the ability to have subicons drawn within the area of an icon. This feature allows the nesting of icons, displaying, for example, a subicon for each employee within a department that is represented by one icon. Subicons may be included to overlay portions of the picture for some or all levels of detail.

The subicon differs from a normal icon in that

(1) if the "parent" icon is moved, the subicons move with it;
(2) if the "parent" icon is erased and its link broken, all of its subicons are erased and their links are broken.

6.5 GDS Operations

The actual construction of a GDS is performed via an interactive graphical editor. This program allows the user or database administrator to select various graphical operations from a menu displayed on the right-hand screen. These operations allow a picture to be created from a set of primitives, such as lines and rectangles, which can be combined with previously defined objects stored in a library. These pictures can be inserted directly into the GDS as data for use in charts and diagrams, or they can be used as templates that will be further processed by the icon-class mechanism described above.

Additional commands allow the database administrator to create new image planes and link them together in various hierarchies and networks. The graphical editor is described in detail in [2].

7. SQUEL, THE QUERY LANGUAGE OF SDMS

Although the usual mechanism for retrieving data from SDMS is the exploration of the GDS, there are occasions when a more conventional symbolic query facility is indicated. This can happen if the user does not know where some particular icon is spatially located, although he can describe it symbolically as, for instance, when trying to find a ship with a particular name. The symbolic query facility is

also used by the database administrator to specify which tuples in a relation are to be displayed as icons in the GDS.

The query language of SDMS is a combination of QUEL, the query language of INGRES, and additions made for the graphical environment of SDMS. The name SQUEL is a contraction of Spatial QUEry Language. Statements from QUEL can be entered directly to SQUEL, as QUEL is a proper subset.

The following commands have been added to the original QUEL commands to produce SQUEL. Each of them accepts a standard QUEL qualification which is used to select particular tuples from a specified relation. The selected tuples are then passed to SDMS, which performs the indicated graphic action. As icons are always associated with tuples, these SQUEL commands can be used to locate and operate upon these graphical representations of information.

(1) **find** ⟨tuple var⟩ **where** ⟨qual⟩. Moves the user to the position occupied by the specified icon. As entire regions can be icons at some level of abstraction, the find command can be used to place the user in the general area of a large group of icons.

(2) **blink** ⟨relation⟩ **where** ⟨qual⟩. Blink finds all the tuples which satisfy ⟨qual⟩. Any icons in the current data surface which correspond to those tuples will blink. They continue to blink until a null blink command is entered or until the user exits the system. The blink command is most useful in identifying which members of a large group of icons meet some specified qualification. After issuing the blink command, the user can move around the GDS, examining the blinking icons in detail.

(3) **frame** ⟨relation⟩ **where** ⟨qual⟩. Similar to *blink* except that the appropriate icons are framed, not blinked. Framing an icon is simply drawing a rectangle around it. Framing is erased similarly to blinking.

(4) **associate** ⟨relation⟩ **using** ⟨icdl⟩ **where** ⟨qual⟩. Creates an icon for each tuple which meets the qualification. The appearance of each icon is determined by the given ICDL. The associate command can be used to create new data surfaces containing only those icons which meet a specified qualification. Thus, while the blink or frame commands allow viewing selected icons in the context of their original data surface, the associate command isolates these icons in their own space.

(5) **change.** Allows the user to update the database through SDMS. It informs SDMS that the user will point to a position within an icon. This signals that the user will update the attribute displayed there. The user then enters the new value for the attribute, and the update is made immediately. This allows the user to update the database without requiring any symbolic specification of the target of the updating operation.

8. SDMS PROTOTYPE IMPLEMENTATION

This section describes the particular prototype of SDMS implemented at Computer Corporation of America. An earlier implementation of SDMS, which utilized manually entered images as opposed to data derived from a symbolic database, was built at the Architecture Machine Group at the Massachusetts Institute of Technology and is described by Donelson [3] and Bolt [1].

8.1 System Environment

The prototype SDMS is written in the C language, running under the Unix operating system on a DEC PDP11/70. The machine has 1.25 megabytes of primary memory and 176 megabytes of moving-head disk storage. Much of the memory is used for storage and manipulation of the bit-map representations of icons and GDSs.

The GDS is viewed via a Lexidata frame buffer display. This display has its own 480 × 640 × 8-bit memory, from which the image on the color television screen is generated.

8.2 Schedule

The SDMS prototype implementation effort has been underway for about two years. The current version of the system provides all of the facilities described in Section 2 for moving about the graphical data surface and generating graphical views of symbolic data.

The current implementation consists of approximately 60,000 lines of code.

9. CONCLUSIONS

The response from visitors to our laboratory who have used the system has been enthusiastic. Only a few seconds are required to learn to operate the controls. During the coming year several prototype systems will be installed, and a more rigorous evaluation will be undertaken, with realistic databases and subjects drawn from the intended user communities.

Two limitations of the SDMS approach are evident at this stage, however, and point up the direction for future work.

First, spatial location of information is best for those problems in which the formulation of a precise query is difficult. The technique encourages browsing through the data, allowing information to be found even when only a vague specification is possible. When a key can be precisely specified, it may be more efficient to type that key directly into the system rather than attempt to recall the spatial location of an icon. More research is needed to determine under what conditions the spatial and symbolic modes are most favored. One would suspect that the ideal combination would allow a mixture of graphical and natural language queries, or would allow the user to manipulate the graphical representations as a means of specifying a query.

A second limitation is the required involvement of the database administrator in setting up the graphical views of the symbolic data. While the level of effort and expertise required is no more than that expended on setting up the data definition language of a conventional DBMS, it would be desirable to have this activity performed in a more automated manner. A major problem is that most existing DBMSs do not contain the notion of an entity, requiring the end user to construct descriptions of real-world objects out of multiple relations or files. This operation is performed in SDMS by the database administrator when he specifies in an icon-class description which relations contain the information to be used in creating a particular icon class. A more general approach would use a higher level query language, such as Daplex [6], that contained such facilities, and a set of

semantic data descriptions that could be used to select among alternative graphical interpretations.

REFERENCES

1. BOLT, R. Spatial data management. DARPA Rep., M.I.T. Architecture Machine Group, Cambridge, Mass., 1979.
2. CARLING, R., KRAMLICH, D., AND FRIEDELL, M. Tech. Rep. CCA-79-25, Computer Corporation of America, Cambridge, Mass., June 1979.
3. DONELSON, W. C. Spatial management of information. ACM SIGGRAPH, 1978.
4. INGRES REFERENCE MANUAL. Memo. ERL-M579, Univ. California, Berkeley, Calif., 1977.
5. MCDONALD, N., AND STONEBRAKER, M. Cupid—The friendly query language. Memo. ERL-M487, Electronics Research Lab., Univ. California, Berkeley, Calif., October 1974.
6. SHIPMAN, D. The functional data model and the data language DAPLEX. Tech. Rep. CCA-79-16, Computer Corporation of America, Cambridge, Mass., April 1979.
7. STONEBRAKER, M.R., HELD, G.D., WONG, E. INGRES—A relational data base system. In Proc. AFIPS, Vol. 44, AFIPS Press, Arlington, Va., 1975, pp. 409–416.
8. SUTHERLAND, I. SKETCHPAD: A man-machine graphics communications system. In Proc. AFIPS 1963 Spring Conf., AFIPS Press, Arlington, Va., pp. 329–346.
9. WELLER, D., AND WILLIAMS, R. Graphic and database support for problem solving. ACM SIGGRAPH 1976 Computer Graphics *10*, (Summer 1976), 183–189.
10. ZLOOF, M. Query by example. In Proc. AFIPS 1975 Spring Conf., AFIPS Press, Arlington, Va., pp. 431–438.

Received October 1979; revised December 1979; accepted March 1980

Vol. 11, No. 2
Sept. 28, 1981
ISSN: 0364-5517

The Seybold Report

Video Display Technologies

What You Can and Can't Do with a Video Display Screen—and Why

UNTIL FAIRLY RECENTLY, most users of publishing systems had little reason to be interested in how characters came to be painted on the screen of their video display terminals. There was this thing which looked like a television tube with simple, monospaced dot matrix characters on it. Granted, some displays were (and are) a lot better than others—better formed characters, a sharper image, better contrast between the character image and the background, more effective anti-glare treatment on the screen, etc. However, the similarities between different displays were much more striking than the differences between them.

In reality, of course, the industry has had somewhat more exotic display technology around for some time in such products as display ad terminals and the Bedford Computer Real Time System. We have been making references to "bit-map displays," "calligraphic displays" and other kinds of display technology in these pages for years, but have never taken the time out to describe exactly what all of this means.

We have finally decided that we cannot put off the discussion any longer. The industry is beginning to see a growing use of video displays which permit the user to see (and manipulate) a variety of type sizes and/or type faces on a display screen—not to mention line artwork, black and white pictures and high-quality color pictures.

Just look, for example, at some of the products which we now discuss in these pages: $17,000 interactive workstations which show the user a representation of true type face and type size on the terminal screen; $400,000 workstations which allow the user to size, rotate and retouch four-color pictures and to perform the most complex kinds of color stripping operations.

The driving force behind this is not that there have yet been any dramatic technological breakthroughs in techniques for painting an image on a screen, but rather that the control electronics which make more exotic displays possible are rapidly becoming faster and cheaper. It is possible to do many things now which simply were not economically practical a few years ago.

This trend is certainly going to continue. The prices of electronic circuitry will continue to drop, and users will want the ability to perform more and more functions on a display screen. Increasingly, the functions which system products can perform are being defined in terms of what the display technology employed makes possible.

It is therefore important for prospective users to understand something about the different display technologies now being employed in graphic arts products, about some of the trade-offs involved with these technologies, and about what changes the future might (or might not) bring. We are not going to tell you whether to buy a Camex or a Xenotron display ad terminal, but we do hope to tell you something about why and how each can do the sorts of things that it does, and what the differences are between this type of product and your text editing terminal.

HIGHLIGHTS IN THE NEWS

Graph Expo drew serious shoppers, in spite of the economy. We bring you up to date with vendor-by-vendor coverage:

Also in *The Latest Word:*

EKC 002002523

Display Technology for the Graphic Arts

The way we work with computers has changed drastically over the last decade. During the sixties, punched cards were the standard medium in most of the computer world, with paper tape also being heavily used in applications involving running text. The user did not get any immediate feedback. Instead, the cards or tape were processed as a batch, and the output of the batch run was examined.

With the rise of timesharing, more interactivity became possible. Many users, each with a Teletype-style printing terminal, could access the same machine. The interactive printing-terminal approach was a great boon for data entry applications, since error checking could be done immediately and the operator could make appropriate corrections.

For text, however, the printing terminal was limited. There was no possibility of browsing through large files, and editing was done via a cumbersome line editor. For graphics, the printing terminal was all but useless (although low-resolution pictures of Snoopy produced by printing line after line of carefully-chosen characters were popular).

The introduction of the cathode ray tube (CRT) terminal changed all that. The CRT offered a flexible display technology, already in mass production (for television), which could be combined with a handful of integrated circuits to produce a cheap, reliable alternative to the printing terminal. Initially, software developed for the printing terminal was used to drive the CRT terminal as well. Hindered at first by the lack of hard copy, the CRT came into its own in text applications when its capability for fluid editing and fast scrolling were realized and software to take advantage of these assets was developed.

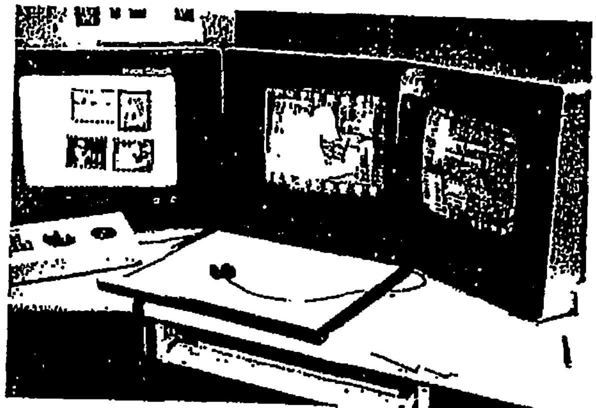
During the period that the CRT was starting to compete with the printing terminal, it was also starting to compete with another type of hard-copy device—the plotter. The plotter was used for engineering and drafting applications and was typically driven from a program that took as data punched cards or paper tape. Plotters tend to be very slow if the plot is a complex one, and the process of getting the data (which was often spacial data—points on a map or a surface) into machine-readable form was tedious.

With the introduction of the CRT into this environment, it became possible to preview a plot quickly so that errors in the input could be corrected. With the addition of a cursor on the screen, plus a light pen or digitizing tablet to make it move, (as well as the proper hardware and software to support this activity) it became possible to enter spacial data into the computer interactively. This, of course, went far beyond what one could do with a plotter.

Thus, in both text-handling and the handling of spacial information, the introduction of the CRT changed the nature of computer processing. It did this by providing new tools for the operator which had not existed in the hard-copy world. In the graphic arts, these two lines of development (text and spacial data-handling) converged in the display-ad terminal and the page make-up terminal.

A third area of the graphic arts is now being similarly altered by the introduction of the CRT, and that is photographic image-handling. The combining of full-color photographic elements into pages is now occurring, and this will be at least as revolutionary for the graphic arts industry as the introduction of the CRT in other areas has been.

The CRT technology is flexible and versatile. That is why it has been able to dominate so many areas. But there are other technologies reaching the market which may offer some advan-



Scitex's Imager console—an impressive example of display electronics. *The full-color image on the center screen can be manipulated in various ways using the digitizing tablet with cursor and the trackball, dial, and 10-key keyboard at the left. The monochrome screen at the right displays status information including the current definitions of the keypad keys. Details are in The Latest Word.*

The details of the way a display (CRT or not) is interfaced to a computer dramatically influence its appropriateness for a given application. Some things are best done by software in the computer, others by hardware associated with the CRT. We will focus on some of the most important choices to be made in these areas.

To start with, we will consider the workings of a very simple terminal, one which is a direct replacement for a printing terminal. In the jargon of the computer business, this is known as a "dumb" terminal.

Anatomy of a 'dumb' terminal

EKC 002002524

Among terminals used for general business data processing, the least common denominator is something called a "dumb" terminal. For reasons which will become clear, the dumb terminal is not generally used for text entry and editing. Nevertheless, the basic functions of the dumb terminal are present in more sophisticated ones as well. An explanation of the workings of the dumb terminal will serve as the basis for a description of more advanced features.

The term "dumb" comes from the fact that no processing of operator input is done at the terminal level. Keyboard input is simply passed through to the host CPU. Characters received from the host CPU are displayed in the next available character position. If a carriage return is received, the following character is placed at the beginning of the next line. If the carriage return is received on the last line of the display, all of the data is moved up a line. This behavior is easily understood if the dumb terminal is viewed as a simple replacement for a printing terminal. The functions are analogous.

If adding a new line at the bottom of the screen forces a line off the top, the data in the top line is lost (at least at the terminal level—the host CPU might still have it stored). Usually, the dumb terminal will also provide a command permitting the CPU to clear the screen and one to allow the positioning of the cursor at any screen location. There are no other facilities. Inserting and deleting of characters or lines, reverse scrolling, and all the other features which are common to editing terminals can only be accomplished by explicit instructions from the host

There are three basic parts in the display-handling part of the terminal: the refresh memory, the character generator, and the CRT itself. The refresh memory stores (as 7-bit ASCII codes) the characters which are currently on the screen. Conventional RAM memory is used, like that in most computers. The refresh memory can hold as many characters as the screen can display —usually about 2000. When a character is received from the host CPU, it is placed in the refresh memory. In a moment we'll return to the mechanism by which these characters, stored as ASCII codes, get converted into dot patterns on the screen. First, though, a word about the cathode ray tube itself.

The raster-scan CRT. In its most common application, as a television display, the electron beam of the CRT sweeps out a series of horizontal lines (called "rasters") as it moves down the screen. Changes in the beam's intensity as it moves along cause changes in the brightness of the screen image. In this way, an image is built up line by line on the screen. This process is repeated over and over with the beam taking one-sixtieth of a second to cover the entire screen and return to its starting point. The process of constantly re-displaying the image on the screen is called "refreshing" the display. The sixtieth of a second that it takes to cover the screen once is called one "refresh cycle." (Television actually uses two complete passes of the screen in each refresh cycle. In Europe, all of this happens only fifty times per second. The principles are the same, however.)

Characteristics of CRT Phosphors

The light-emitting coating on the inside of the front surface of the CRT is called the *phosphor*. A phosphor is a substance which gives off light when bombarded by electrons. A number of substances share this property, but they differ in other characteristics.

Color. Such a characteristic is the *color* of the phosphorescence. A wide variety of colors is available. For color displays, the phosphor colors have to be chosen in such a way as to make possible the display of as much as possible of the visible spectrum. Unfortunately, there are some colors which existing phosphors cannot duplicate. For monochrome displays, white or light green phosphors are most often used. Orange phosphors are also used, particularly in Europe.

Persistence. Another important characteristic is persistence. This is a measure of the rate at which the phosphoresence dies out after the electron beam has been removed. It is usually given as the time required for the phosphorescence to drop to 10% of its original brightness. For editing terminals, the persistence should be less than the period between refreshes, so that moving items don't appear to "smear" on the screen. P4, the phosphor used in black and white TV and some VDT's, decays to 10% brightness in about 60 microseconds. P31, a common green phosphor for displays, takes 40 microseconds. These phosphors fall into the "medium short" persistence categories, and while shorter-persistence phosphors exist, they are not generally used in displays. Other phosphors available for display use have persistences ranging all the way up to 10 seconds (for P26). Most of these are better suited to applications like radar displays than to the graphic arts.

Efficiency. Efficiency is a measure of the amount of light produced by a given beam energy. This is not a major factor in CRT's for direct viewing, but it is important in CRT typesetters, whose imaging speed is in many cases limited by their light output.

The key factor in determining the rate at which refreshing must occur is a psychophysical phenomenon called "flicker fusion." When an image is displayed at a rate of less than 30 presentations ("frames") per second, most people begin to notice the individual images as flicker. When it is presented 60 or more times per second, the images seem to "fuse" and most people see one continuous image. With the phosphors typically used in terminals, a given dot is actually only "on" for a fraction of a thousandth of a second after the electron beam excites it. At 60 frames per second, it will be about 17 thousandths of a second before the beam comes back again. Thus, the dot is actually dark most of the time but it appears continuously bright due to flicker fusion. Long-persistance phosphors are available which can be refreshed at lower rates without producing flicker. These are not suitable for the display of moving objects, however, since the persistence of the phosphor will cause a "smearing" effect behind the item being moved. Scrolling text, for example, would appear smeared and could be difficult to read.

Now we return to the dumb terminal. Like all raster displays, it always scans the screen in the same sequence and and at the same rate. The only variable is beam intensity, and by varying this, a dot or horizontal row of dots can be produced at any point along any raster. Vertical lines can be built up by placing dots underneath each other on successive rasters.

When characters are displayed on the dumb terminal, the pattern of dots corresponding to the characters has to be recreated on the screen with each refresh cycle, sixty times a second. The terminal must have a mechanism for constantly repeating the carefully-timed sequence of "on" and "off" signals for the electron beam. This requires a memory of some type.

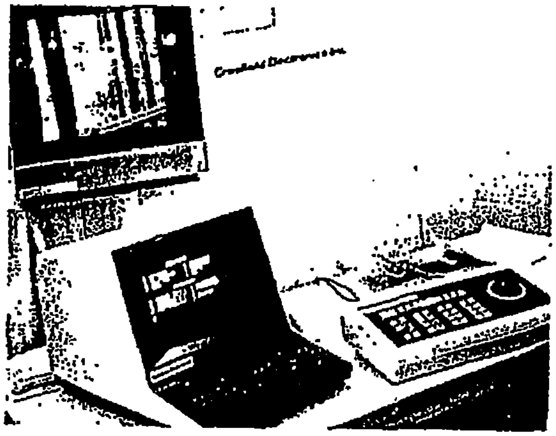
The bit-map approach. One way of arranging this memory is to have one bit of memory for each dot location on the screen. The memory is sequentially scanned synchronously with the scanning of the electron beam. If a "1" bit is found in a given memory location, the beam turns on for an instant (creating a dot); if the bit is a "0" the beam stays off. This arrangement is known as a "bit-map." There is a one-to-one correspondence between bits in memory and a dot locations on the screen.

The bit-map approach is used for some applications, including several in the graphic arts, but not for our dumb terminal. To represent every dot position in a 2,000-character display with each character requiring 48 dots (5 wide, 7 high, plus a row of dots at the side and bottom to separate adjacent characters) at the very minimum, would require 96,000 bits of memory. This is a relatively expensive approach, not practical in a low-cost terminal. (The electronic design of a basic bit-map terminal is very simple, however.)

The character generator. Instead of one bit per dot, the dumb terminal stores an ASCII code for each character position. This is only seven bits per character instead of at least 48. (In many terminals there will be more than seven. A few extra bits per character may be used for indicating bold, blink, etc.) The tradeoff for the savings of memory is that additional hardware is needed to convert the ASCII code into a pattern of dots during the refresh cycle.

The dot-producing hardware is called a "character generator." It works by using what amounts to a table-lookup procedure. Let's suppose the characters are five dots wide and seven high, and 80 of them appear on one line. For each character the read-only memory of the character generator contains seven five-bit strings, one for each row of dots in the character. There are "1" bits for dots that need to be turned on and "0" bits for the rest.

The character generator takes as input the ASCII code of the character to be displayed next and the raster number (i.e.,



Crosfield. *The two extremes in display sophistication are represented in Crosfield's color retouching station. In the foreground, a standard alphanumeric "dumb" terminal displays status information and operator commands. In the background, a color display of the image on which the operator is working is driven by lots of dedicated high-speed image-processing hardware.*

which of the seven rows of dots is the current one). It will take seven rasters to display a full line of characters on the tube (plus an eighth raster to separate this line of characters from the line below). The first 80 characters of refresh memory will appear on the first line. These 80 characters are read from memory seven times in a row, concurrently with the scanning of the first seven rasters by the electron beam. Each character's ASCII value is fed to the character generator on each of the seven passes. On each pass, a raster counter is increased by one to indicate the number of the current raster. After seven rasters, the count goes back to one and the process repeats with the next 80 characters (representing the next screen line).

The character generator proceeds as follows: as it receives the ASCII value of the next character from the refresh memory and the raster number from the raster counter, it looks up the corresponding five bits in its table. These are then fed sequentially to the electron beam circuitry, resulting in dots on the screen.

Although the explanation we have given may seem complex, the actual circuit is a simple and inexpensive one. A given character may appear many places on the screen, but its dot pattern is only stored once, in the read-only memory of the character generator. The conversion of the character into dots takes place "on the fly" during refresh, which is why this approach is so efficient in its use of memory. This method of generating characters is almost universal in the data processing world. Note, however, that certain types of flexibility have been lost. All characters have to be the same size and they can appear only in pre-determined regularly-spaced locations. Proportional spacing is not possible. In this basic form, there is no provision for reverse scrolling, horizontal scrolling, or any other editing niceties which involve relocating text on the display.

Intelligent vs. dumb terminals. Intelligent terminals are terminals with some local processing capabilities (frequently based on a microprocessor). Whereas dumb terminals always use display techniques similar to those just described, intelligent terminals incorporate various display schemes. We will use the dumb terminal as an example of a simple refresh-memory-plus-character-generator system, but the reader must realize that the same display technology is used in some intelligent terminals, and some terminals without local intelligence use different write "dumb terminal" than to write "simple-refresh-memory-plus-character-generator."

In a similar way, we discuss the terminal and its "host CPU" as if these were separate entities; as if certain parts of the display system (i.e., the refresh memory and the character generator) were part of the terminal, while the "host CPU" was not. In actual practice, the physical locations vary. For example, the character generator for a given terminal may be inside the terminal itself, inside a cluster controller serving several teminals, or inside the main CPU cabinet of the entire system. The "host CPU" may be a microprocessor in the terminal itself or a mainframe miles away. These differences, while important in other contexts, are not relevant here. For clarity, we will stick with our initial terminology, with the understanding that no implications about system architecture are intended.

Sophisticated character generators

The simple table look-up scheme for character-generation outlined previously is efficent, but limited. It can be extended in a number of ways to make it more useful. It can be extended to accomodate additional fonts or special characters. Its output can be further processed to provide different intensity levels, reverse video, and blink.

Fast scrolling can be provided if the refresh memory is enlarged to the point where a substantial amount of text (beyond that currently displayed) is stored, and if the character generator can access not just one specific area of this memory, but any block beginning at a point designated by the host CPU. With this facility, the contents of the screen will be the 2000 or so characters which begin with the memory location specified. Then scrolling becomes (from the CPU's point of view) simply a matter of sending the addresses of successive line-beginning characters to the terminal. These will appear in the first character position on the screen, and the rest of the screenful is handled automatically by the character-generating hardware.

A display with the features just outlined can be the basis for an efficient text entry and editing system. Most terminals used for these purposes have most of the elements described.

There is another feature one occasionally sees in terminals for this application which adds some flexibility. That feature is user-definable character shapes. In the standard character-generation circuit, the characters are stored in read-only memory. But if standard read/write memory is used, new character shapes can be loaded.

Alphatype offers this feature on the MultiSet system. A character-creation facility is also provided so that the user can design new characters needed for a particular job. And the newly-designed character can be stored on disk for later use. This capability is valuable for math, foreign languages, and other work involving special characters.

Variable spacing and sizing. For text entry and editing, single-size monospaced displays are adequate. But for some applications (e.g., display ad terminals) variable sizing and proportional spacing are needed. Several approaches are possible to achieve this. One is the use of the same basic approach (refresh memory feeding a character generator) but with a more sophisticated character-generation scheme.

Proportional spacing can be handled by the standard character-generation scheme with one addition. Besides storing the dot pattern for a character, the read-only memory would store the character's width. It would be stored as a count of how many dots wide the character is. The character generator would then read that number of dots out and then go on to the next character. This approach, while workable, does not permit

Raster Displays: Bandwidth and Other Esoterica

The most common raster-scan CRT device is the television. A television image consists of 525 rasters, of which half are displayed during a given one-sixtieth-second "frame," the other half during the succeeding frame. Of the 525 rasters, only about 350 are actually visible on the face of the television. (In Europe different numbers apply)

The display of every second raster in one frame and the balance in the next frame is called *interlacing*. This is standard practice for television and it is quite satisfactory with moving images. But with stationary images (particularly narrow horizontal lines) it can lead to flicker. For this reason, displays for the graphic arts are not usually interlaced. All rasters are displayed in each frame.

Bandwidth is a measure of how fast the CRT's electron beam can be turned on and off. It is closely related to the potential number of dots per raster. In North American television, each horizontal sweep of the electron beam takes about 53 microseconds to cross the display. (It requires 10 microseconds more to return, for a total time of 63 microseconds per raster). The bandwidth of a standard TV signal is around 4 MHz, which is equivalent to about eight on-off transitions per microsecond. Thus, in one 53-microsecond raster, about 424 pixels (dot locations) can be displayed.

The bandwidth affects the number of charactrs per line. If the characters are six dots wide (five dots wide plus one dot to separate characters—the absolute minimum), then about 70 of them can fit on one line of 424 dots. If they are eight dots wide (as is commonly the case) then only 53 will fit. The number of dots per line can be increased by increasing the bandwidth.

Non-interlaced displays operating at standard video rates can display only about half the number of rasters that the interlaced display can (*i.e.*, about 175 lines). For characters eight dots tall (the minimum for legibility), this implies a capacity of 22 lines of characters. This is reduced to 17 lines if the characters are a more reasonable 10 dots tall.

To display more lines of characters, more rasters are needed. To generate more rasters, the electron beam has to be moved faster across each one, since the total time to cover the screen (a sixtieth of a second) cannot be reduced without introducing flicker. Doubling the speed of the beam doubles the number of rasters which can be displayed, but halves the time per raster. This causes the number of dots per raster to be halved as well. If the bandwidth is not increased. Increasing the bandwidth permits more rasters and/or more dots per raster, but the cost of displays with high bandwidths is much greater than the standard TV variety.

Several additional factors enter the calculation when color displays are considered. Color displays have three electron beams all moving across their rasters together. A *shadow mask* ensures that the "red" beam strikes only the red phosphor areas, and similarly for the green and blue. The physical size of the openings in the shadow mask dictate the spacing between phosphor dots or stripes. In standard TV's, there are at most a few hundred phosphor locations per color per raster. This sets a physical limit on dot spacing. With specially-constructed color monitors, this limit can be pushed to nearly 1000 dots per line, but at great expense.

The preceding discussion of color applies when separate signals drive the red, green, and blue beams and each signal has the same bandwidth. This is not the way television works, however. The standard North American color TV signal consists of a "luminance" (brightness) signal on which a "chrominance" (color) signal is superimposed. Monochrome receivers display only the luminance information. The chrominance signal has less than 1 MHz of bandwidth—far less than the luminance. The result is that color changes have to be much more gradual than changes of intensity. If an attempt is made to display dot-matrix characters of one color against a different background color the dots have to be large. Otherwise, "fringes" of spurious color form at the edges of the characters. (This is why personal computers which connect to the home television have so few characters per line.)

that we know of. It is used in the word-processing industry, however.

Sizing is more difficult. If only a few sizes are required, a read-only memory can be provided for each size. This would not handle the usual requirements for setting ads, however, since many sizes might be used. One way to get around this problem would be to use a read/write (that is, RAM-based) character generator which is loaded via a sizing procedure. The characters are stored at a master size, but sized appropriately when they are loaded into the character generator. The sizing operation must also figure out width and height information (how many dots horizontally and how many scan lines vertically the new character will occupy). A variation on this approach is used in CRT typesetters, but not in any terminals that we know of. The Xerox Star, unique among interactive terminals in that it displays "true" typefaces, employs a related scheme: all the characters of each font are stored in each available size on disk at the terminal. As they are needed, they are loaded into memory. This approach is flexible but very memory-intensive. It could be used for display ad terminals if the number of possible fonts and sizes were not so vast and the difficulties of digitizing fonts for a relatively coarse display were not so severe.

The bit-map display

A different approach, which is found in display ad terminals (as well as in typesetter preview displays), involves refreshing the screen from a bit-map memory. The bit-map has one bit for each dot on the screen, permitting very simple refresh electronics. The host CPU can turn dots on or off by changing bits in the bit-map.

Generation of characters on a bit-map display can proceed in a somewhat more leisurely fashion, since it doesn't have to occur in lock-step with the raster movements of the CRT beam. Nevertheless, it must occur fast enough for interactivity. In particular, when blocks of copy are moved around by the operator, a great deal of character generation activity is occurring. The image of each character has to be erased from the bit-map at its old location and redisplayed somewhere else.

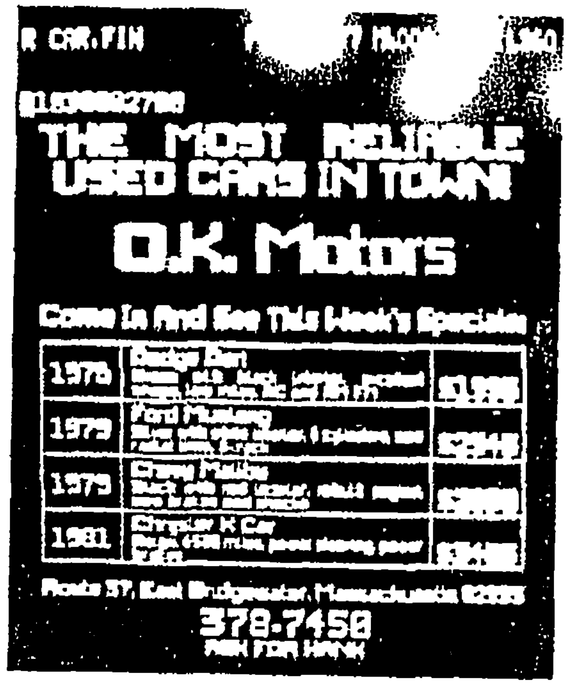
The display on the Bedford system appears to operate in this fashion. This system is designed to cope with flowing text and the display does not appear to have any problem keeping up with the rate at which the host computer can format the text to be displayed.

However, for an application such as a display ad terminal where reformatting is not a limiting factor, the slowness of the character-generating process could become objectionable. Instantaneous response, especially to dynamic activities such as the movement of the cursor or a block of copy, is highly desirable. To move a block of copy smoothly across the screen requires a character-generation scheme that can update the block's location in a tenth of a second or less. While there is no theoretical reason that this cannot be done on a bit map display, current devices in use in the graphic arts can't do it.

EKC 002002527

A hardware solution to the instantaneous-movement problem is to add a second bit-map just for the copy which is being moved. The image created by this bit-map is combined with the image created by the main bit-map as refresh occurs. The copy in the secondary bit-map can be made to appear to move relative to that in the main bit-map by timing tricks. For example, if scanning of each raster of the secondary bit-map begins after 20 bits have already been read from the main bit-map, the image of the secondary bit-map will appear shifted 20 dots to the right relative to the main map. Similarly, if 12 rasters are generated from the main bit-map before beginning the scanning of the secondary one, the effect will be to shift the image of the secondary bit-map twelve dots downward. Such timing is readily controlled by software-loaded timers, making it easy to implement. The problem with the approach is the doubling of memory which it requires. Memory has been a major cost item in a bit-map display, although this is changing as memory prices continue to decline. It is not used in display ad terminals at present. This type of approach is used in color page make-up systems, and we will come back to it in that context later. It is also used as a cursor-generating method by the Xerox Star. The Star's cursor is a 16 by 16 bit array which is freely and continuously movable around the screen and whose contents can change under software control. Thus the cursor could appear as an arrow in one context or one part of the screen and as a box in another.

In the Raytheon, Xenotron, Mycro-Tek and Linotype display ad terminals a cursor is used to indicate where a block of text is to be moved to, and only the cursor is freely movable. The selected block of text stays in place until the operator positions the cursor and calls for the move. Then the block is erased and redisplayed in the new location. This reduces the problem of interactive movement down to the problem of cursor movement. The cursor is most easily handled if it is not written into the bit-map at all but is generated by a simple circuit whose output is combined with the bit-map refresh output. A novel cursor approach is that of Xenotron, whose block cursor assumes the shape of the block of text being transferred. If the user decides to adjust the point size or line length within the block during the block positioning process, the block will change its shape to reflect these changes. Once the user is satisfied, the block is removed and the actual text is generated in place. The beauty of this approach is that it gives a fair amount of valuable size and position information without constantly regenerating the text. Generation of a solid block is very much simpler. MycroTek's display ad terminal uses a similar approach.



Bit-map display of the Xenotron Video Composer. *This one was in Itek's booth at Graph Expo.*

Instantaneous Response vs. Refresh-Rate Processing

There are two questions about human visual perception which have fundamental importance for display technology. The first is : "How long an interval can occur between presentations of an image before flicker becomes perceptible?" The answer is, about a sixtieth of a second. This determines the refresh rate.

The second question is: "How long an interval can pass between the issuing of a command and its implementation on the display if the change is to be perceived as 'instantaneous'?" The answer is, about a tenth of a second. Thus, even though several refresh cycles may occur between an operator action and the display's response to it, the response will be perceived as instantaneous if it occurs within about a tenth of a second.

Since the rate of processing required for refreshing is not too different from the rate required for instantaneous response, display designers often incorporate the hardware needed for the one into the hardware needed for the other and do everything at the faster (refresh) rate. This is a matter of design convenience. However, it has led to some confusion of the two concepts.

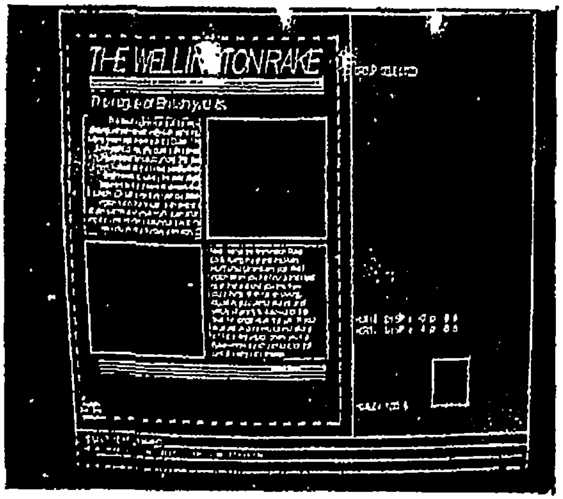
Vector display CRT's

In applications like computer-assisted engineering design and circuit board layout, the type of display most often used is a vector display. In the graphic arts, the vector display is most commonly associated with display ad terminals. It is used in Compugraphic's AdVantage and Camex's display ad terminals, which together account for a majority of all display ad terminals in use. It is also used on the Imlac system and in some page make-up schemes.

The vector display draws line-segments on the screen between points specified by the host CPU. Unlike the electron beam of the raster display (which moves in a fixed pattern) the beam in the vector display can move in any direction. Like the raster display, the vector display must continually refresh its screen at a fast enough rate to avoid flicker. To do this, it retraces each vector on the screen at least 30 times a second.

The vector display has a memory containing all the information needed to reconstruct the page. This includes not only the characters to be displayed (including size and position information) but also the general parameters currently in effect, such as zoom and image position relative to the absolute screen coordinates. All of this data is processed repeatedly, once for every refresh cycle. The display processor outputs endpoint values to a vector-generation circuit which draws them on the screen.

Characters are represented on the screen as a series of



Compugraphic's AdVantage display ad terminal at Graph Expo. The AdVantage uses vector display technology.

might be represented by a single vector. More complicated characters require more vectors—two for a "v," four for a "w," etc. The number of vectors is important, since only a limited number can appear on the screen at one time. This is because the refresh process must "draw" each vector in turn, and time is consumed by this process. If the number of vectors to be displayed is too large to be drawn in a thirtieth of a second, flicker will occur. The number of vectors possible is usually in the tens of thousands, thus making it possible to display several thousand characters without flicker. When characters are drawn as vectors, the size of the characters doesn't affect the timing much. The basic limit has to do with the number of vectors, and a given character is generally represented by the same number of vectors regardless of its size.

Further refinements in the handling of the display memory are possible. For example, during the refresh cycle the display list data can pass through a high-speed arithmetic processor before being sent to the screen. This processor can be instructed to add a constant to (or subtract it from) all the x-coordinates, causing the image to move to the right (or left). Similar manipulation of y-coordinates causes vertical motion. Multiplication by an appropriate constant creates an enlarged or reduced image. By giving the operator a stylus or joystick to control these parameters, continuous zooming and scrolling in all directions may be achieved. (While similar scrolling capability is reasonably easy to design into a raster terminal, continuous zooming is not.) The accuracy of the vector display is also better than the raster display, since the latter can only produce dots on the rasters while the former can position them arbitrarily. A related observation is that, because of the dot-by-dot nature of the raster display, lines which are not vertical or horizontal have a stair-step appearance. This effect, known as the "jaggies," is not present in vectors on a vector display.

If the strong suits of the vector terminal are accuracy and interactivity, then its weaknesses are limited capacity, inability to display solid areas or photographic imagery, and its minimal color capabilities.

The capacity problem has already been touched on. There is a limit to the number of vectors which can be displayed in a refresh cycle, and in text use this occurs when a few thousand characters are on the screen. Solid areas would in theory be representable by many vectors side by side, but in practice this cannot be done because of the restriction on the number of vectors. A similar argument applies to the representation of photographic material.

Color displays, though available, are of limited application since they can only show lines on a dark background, not areas of color. The most frequently-used vector color technologies provide a very limited range of colors. They have not found use in the graphic arts thus far.

To understand the strength of the vector terminal in the display ad market, it is necessary to recall the notion of "instantaneous response." In the composition of display ads, the operator is making constant aesthetic judgments as the various elements are sized and positioned on the screen. To make this process a smooth and continuous one, all of the common operations involved in positioning and sizing must be instantaneous. In most of today's raster-based display ad terminals, only the cursor exhibits this type of instantaneous response. Some also offer continuous movement of the entire image in both axes (to permit working with images that won't fit on the screen) and instantaneous (but not continuous) zoom.

The vector-based terminals, on the other hand, offer many more instantaneous operations, including sizing, dragging of copy blocks around the screen, and continuous zooming. There is no inherent reason why such facilities cannot be incorporated into terminals with raster displays—they are in fact incorporated into some of the raster-based color page-makeup systems already. Doing so is much easier with vector displays, however, and in the display ad field, only the vector-based systems have offerred them so far.

To get from raster displays the responsiveness usually associated with vector displays requires special hardware. This has usually meant specially-designed boards of medium-scale integrated circuits, but recently chips have been developed specifically for this application. One such chip, National Semiconductor's 7220 "Graphics Display Controller", can store lines and arcs into a bit-map memory at a rate of over a million bits per second, which is equivalent to changing about 20,000 bits per refresh cycle. For some applications, faster speeds are possible by combining several 7220's. Specially-designed integrated-circuits of this sort should help the raster-based display to become more competitive in applications like the display ad terminal.

The storage tube

A third type of CRT that is used in some graphic arts applications is the storage tube. The storage tube may draw images either as rasters or vectors (the latter is the usual approach). Once they are drawn they do not fade. This is because, in addition to the writing beam, there is a stream of unfocussed "flood" electrons which causes any region of phosphor which is excited to remain in the excited state. Thus, an area which is "turned on" remains bright until the flood electrons are turned off, at which time the entire screen is erased. The result is that storage tubes do not require refreshing and therefore no refresh memory is needed. Character generation can be completely by software, without the need for a bit-map.

The negative side of the storage tube is that images cannot be moved or deleted without clearing the screen and then regenerating it. Not only is this a slow process if the image is complex, but it also requires that the image be stored somewhere in the host computer's memory for reconstruction purposes. This negates the main advantage of the storage tube—that memory is not required. Thus the storage tube is only used where updating is relatively infrequent and where material is added far more often than it is moved or deleted.

In the graphic arts, storage tubes are used primarily as typesetter preview devices. They are also occasionally used for page make-up terminals (for example, Crosfield's monochrome "planning station" display is a storage tube).

EKC 002002529

CRT's, Health, and Standards

In recent years there has been an undercurrent of concern among users of CRT's that there might be health hazards involved in their use. At first, the focus was on the possibility of x-ray emission. This possibility has been extensively checked in the field and health-threatening levels of x-radiation have not been found to date.

Still, there are higher numbers of health complaints among users of CRT terminals than among non-users. This has led investigators to pursue three other possible types of problems: ergonomic and environmental ones (glare, postural considerations, keyboard height, etc.); psychosocial ones (terminals as job-threatening or job-stultifying elements); and unknown physical ones (subtle effects of very high flicker rates, electromagnetic effects of some kind, etc.).

Ergonomic and environmental issues are the easiest ones to do something about. In Europe, standards have been widely discussed and some have been adopted into law. Among the display-related matters considered are color, contrast, luminance, size, spacing, and polarity (light-on-dark vs. dark-on-light) of the displayed characters, height of the display, ambient light levels, and others.

Psychosocial matters related to the introduction of the CRT terminal were brought into focus by a study by the National Institute of Occupational Safety and Health published earlier this year. This study involved 250 VDT operators and 150 controls. The VDT operators were about evenly divided between "professionals" and "clerical workers."

The "professionals" studied were all part of a large newspaper's editorial and production staff. The control group in the study was clerical workers working with paper documents.

The level of health complaints of the clerical and professional workers were quite different. Among the major complaints of the clerical workers were: stiff or sore wrists, changes in color perception, hand cramps and loss of feeling in wrists or fingers. All of these were reported with at least three times the frequency of the control group. The following were at least twice as common among clerical VDT users as among the controls: fainting, blurred vision, swollen muscles and joints, numbness, neck/shoulder pain, and loss of strength in arms or hands.

The situation among the professional users was quite different. They did not report health problems at anywhere near the levels of the clerical users. Only three complaints were above the levels of the (non-user clerical) controls: burning eyes (36% above controls), eye strain (30% above), and irritability (21% above).

The report of the study emphasizes the importance of job perception and perception of the role of the terminal as a helpful tool on the one hand or a monitoring one on the other. "When the job features of the various groups are examined," the report says, "we see that the clerical VDT operators held jobs involving rigid work procedures with high production standards, constant pressure for performance, very little operator control over job tasks, and little identification with and satisfaction from the end-product of their work activity. In contrast to the clerical VDT operators, the professionals using VDT's held jobs that allowed for flexibility, control over job tasks, utilization of their education and a great deal of satisfaction and pride in their end-product. While both jobs had tight deadline requirements, the professional operators had a great deal of control over how these would be met. In their case, the VDT was a tool that could be used for enhancing their end-product, while for the clerical VDT operators, the VDT was part of a new technology that took more and more meaning out of their work." The fundamental point here is that VDT's are just one aspect of a job situation and it is difficult to separate complaints due to physical characteristics of VDT's from those due to other job features associated with the automation of jobs.

We obviously don't know if there are yet-to-be discovered physical effects of CRT use. The widespread use of CRT's is still too new to be absolutely certain about this. But one minor side benefit of the alternative display technologies discussed in this article is that they will be able to serve as a basis for comparison with the CRT in studies of these issues. The alternative display technologies vary widely in their characteristics: spectrum of light emitted, flicker characteristics, size, viewability under various light conditions, etc. For any given characteristic which CRT's possess, there is a display technology which offers an alternative to it.

Terminals based on these alternative technologies will make it possible to separate out the effects which are associated with the cathode ray tube and those (like postural and psychological effects) which are by-products of the introduction of terminals independent of display technology. If there are indeed health problems associated with the technology of the CRT itself, this will permit them to be isolated. If there are not, this will allow the real causes of the health complaints to be identified and corrected.

Image-handling displays

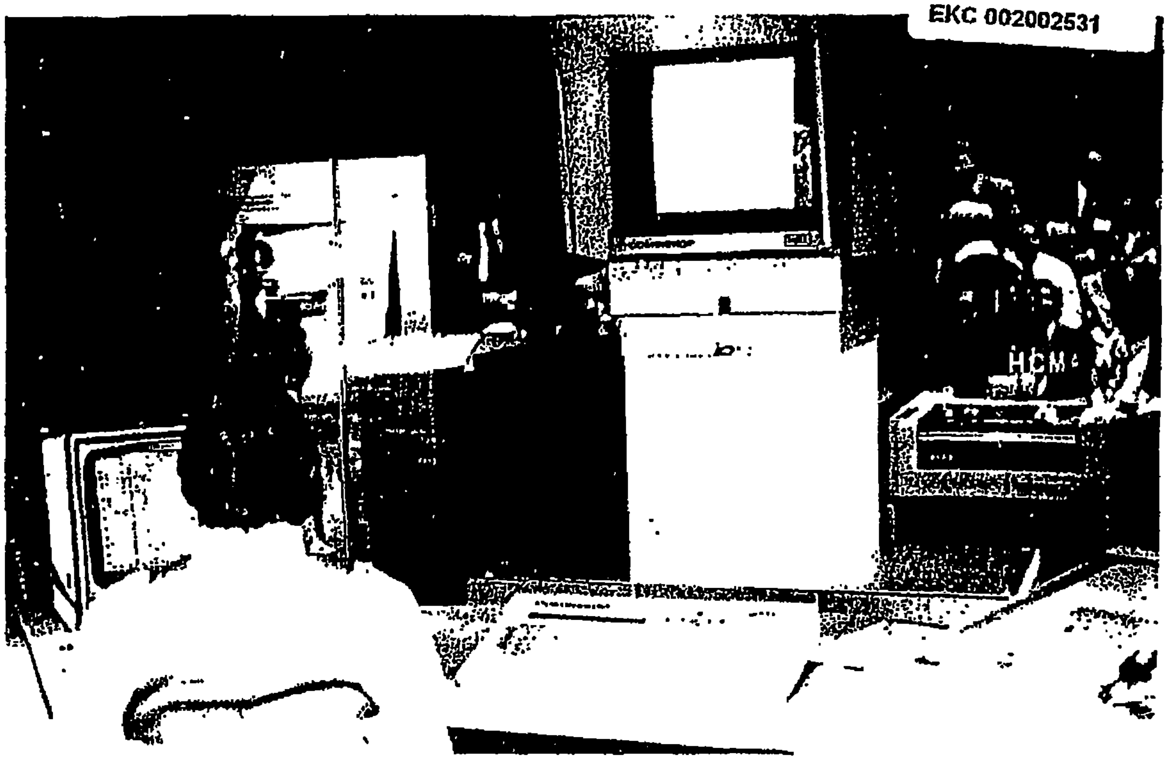
Displays for handling photographic imagery are used for page make-up systems and preview terminals for color scanners. These displays differ in several respects from those discussed so far. And yet, many of the underlying principles are exactly the same. All image-handling displays used in the graphic arts are raster-scan CRT's.

The basic design of the image display is like that of the bit-map display discussed previously. But for flexible image manipulation, the image display must go far beyond the capabilities of the bit-map device. The terminology changes, too. Where we spoke of "dots" in the context of bit-map displays, we will now speak of picture elements or "pixels." The memory used for refreshing the screen was called the "bit-map;" now it becomes the "frame buffer," a term borrowed from the world of video. In dling full color work. Monochrome image systems are very similar, except that less data per pixel is required. Most of the discussion could be applied to either.

The most obvious additional requirement for handling imagery is color information. Instead of a single bit per pixel, there will have to be a fairly large number, typically 24 (eight bits of intensity data each for red, green, and blue). This is going to require vastly increased memory. The CRT display has three electron beams, all scanning their rasters in concert. The intensity of each beam has to be separately controlled. Mere on/off information will not do—this has to be analog information capable of representing a continuum of intensities. There will therefore have to be three digital-to-analog converters, one for each color.

The amounts of memory involved in running a display of this type are staggering. Suppose there are 500 pixels in each

EKC 002002531



Hell's Chromacom color image manipulation system. *The display at the upper right shows the image being worked on. A wide variety of operations can be performed on the image via special-purpose electronics. These operations are accessed via the keyboard below the display. The alphanumeric display at left (behind operator) is used for status and file-handling information.*

screen. Each represents 24 bits (3 bytes) of color data. Therefore, refreshing the screen requires three-quarters of a million bytes of memory. And as we will see, most systems for the graphic arts will need far more than this.

Working with this quantity of memory is awkward. Any sequential operations performed on the data in it will be very time consuming. Consider, for example, adjusting the color values of an image. Ideally, the operator should be able to adjust a set of three dials or sliders (say, one for each color) and see the color change on the screen. The system should be quick enough so that the change on the screen keeps up with the operator's movement.

Suppose the operator turned the "red" dial to a position that meant that 10% more red was needed. To accomplish this change via a computer program would entail reading the current position of the dial and running through the screen memory, reading the current value of the red data, increasing it by 10%, and storing it back. One such pass through the screen memory with a reasonably fast computer would take at least a couple of seconds. To keep up with the operator, on the other hand, several passes per second are necessary. Any other manipulation that affects a large fraction of the screen pixels is equally slow, and some are much worse.

The point of this analysis is that memory requirements of this magnitude, combined with the need for sub-second response times for changes, cannot be handled by conventional computers and programs. Special hardware, designed specifically for the purpose, is required. Such hardware is found in image-processing systems for the graphic arts.

There are two places where this hardware can be used. Recall the difference between the dumb terminal, with its character generator, and the bit-map display. Characters to be displayed on the bit-map device are converted into dots once, after which those dots remain stored (and displayed) until they are changed. In the dumb terminal, on the other hand, characters are converted into dots every sixtieth of a second as the screen is refreshed. In the bit-map case, character-generation is only performed once, and its speed is not critical. In the dumb-terminal case, character-generation is performed again and again at a speed which is precisely synchronized with the scanning of the electron beam. This can be described as processing at "video" rates. The precise speed of video-rate processing is a function of the number of pixels and the refresh rate. If there are a quarter-million pixels on the screen and the screen is refreshed 60 times a second, then the video processing rate is 15 million pixels per second.

Just as character generation can be done either at video rates in the refresh process or at slower rates in the course of updating a bit-map, the same two possibilities exist for processing image data. Video-rate processing calls for special-purpose hardware but is very fast (any change is in effect as of the next refresh cycle). Updating the frame-buffer is often much slower but can be done by general-purpose computers (or special-purpose circuits which aren't able to keep up with video rates). One important point about processing at video rates as part of the refresh cycle is that the data in the frame buffer itself need not be changed. The processing occurs after the data has been read out of the frame buffer. The processing hardware is a part of the data path from the frame buffer to the CRT. This means that the changes made to the data can be kept separate from the data itself (which need not be changed). This is a very useful feature in some situations.

"Instantaneous" response in a color image-manipulation system inevitably requires special hardware. General-purpose computers can't process the huge amounts of data in a reasonable period of time. The possibility of processing the frame-

buffer output at video rates has already been mentioned. If a feedback path is provided permitting the flow of displayed data back to memory, then the processed image can be made permanent once the operator is satisfied with it.

Processes which occur at less than video speeds can appear instantaneous if they occur within about a tenth of a second, and some operations are easier to implement at this speed than at full video speeds. This type of processing generally affects the image indirectly through changes to the frame-buffer. Some operations in this speed range are also implemented by several loops through the video processing hardware, returning each successive version of the image to the frame-buffer memory via a feedback path.

An attractive facility for the graphic arts is the ability to incorporate one image into another (the screen equivalent of paste-up). The relative positioning of the two images presents a technical problem, however. Ideally, the operator would be able to "push" the second image into position on the screen while the elements which had previously been positioned remain still. The counter mechanism described in connection with bit-map displays is no help in this situation because it would cause everything, including the previously-positioned images, to move. The only real solution is to have two frame buffers, the contents of which are combined during refresh. Each can have a separate counter for movements relative to the screen (and to each other). The implications of this solution are a bit awesome, however: double the refresh memory is now needed. (Readers who watched the Scitex system in action at Print '80 will recall that the positioning of a new image on a page that is being made up was performed by moving a cartoon-like representation of the new image. This approach avoided the necessity of a full second frame buffer since only a few bits per pixel were used. At Graph Expo this year, Scitex demonstrated the same operations with the full-color image. The Hell system also gives the user full detail in the second image.)

With two frame buffers, each 24 bits "deep," we now have 48 bits per pixel and we're not through yet. (It's not difficult to see why memory prices are the key to equipment prices in this field.) These systems don't generally use more than two frame buffers (although if that were feasible it would be very useful) but there are a number of other uses to which memory can be put in this context. Line art, for example, is usefully stored separate from the full-color imagery. The color page make-up systems all have rule and border-generating facilities. Some of them can also generate Benday tints, etc. Generally, there will be a large number of pixels generated, all of which have the exact same color. To store 24 bits for each point in this situation is wasteful, especially during the creation and positioning stage, when the line art is still being altered and checked. Instead, it is useful to have a screen-size bit-map for this purpose which is just one bit deep but which can be assigned any given color. All the "1" bits would then turn up as pixels of that color on the screen. System-generated masks, with the background dropped out according to the color value of the pixels, can likewise be readily handled with such a bit-map. (A screen-sized bit-map one bit deep is generally called an image "plane.")

These are just the basics of what is available in terms of hardware for image processing. Special hardware is available for other operations like image rotation, image contrast enhancement, airbrushing effects, zooming, and many others. Some of these processes are similar to those already described, and others are quite different. Image processing is a fast-moving field whose application areas are growing rapidly as memory costs come down. The technology is exciting and well-suited to many areas of the graphic arts—not just in production processes but also (as prices continue to drop) in creative areas as

Flat panel technologies

Light-emitting diode displays. Light-emitting diode (LED) displays are the ubiquitous red displays found on calculators, watches, cash registers, digital clocks, etc. They are also available in green and yellow. (A number of companies are working on blue LEDs. If these could be economically produced, full color displays would be possible.) LED's have appeared in the graphic arts market only as single-line displays on some direct-entry typesetters and as indicators, counters, and the like on typesetters, communications equipment, etc. Some electronic typewriters use a single-line LED display.

Although large arrays of LED's (of a size suitable for an editing terminal or for graphics) have been produced, the cost of fabrication is very high (though in volume it would be reduced) and the power consumption is extemely high. LED's have some advantages over other solid-state display technologies. The fact that they behave as diodes makes the wiring for an X-Y array extremely simple. They can be driven from the low voltages of common digital integrated circuits. The power requirements are so high that they would seem to negate these advantages, however. It is the high power consumption which is likely to limit the future graphic arts use of this technology.

Liquid crystal displays (LCD's). Liquid crystals, like LED's, are generally associated with watches and calculators. Unlike LED's, they produce no light themselves. The liquid crystal can be switched between two states, one which permits light to pass through and one which does not. If a reflective surface is placed behind the liquid crystal, ambient light is reflected back to the observer from the transmissive areas of the display. The non-transmissive areas do not allow the light to be reflected and appear dark. Alternatively, backlighting can be used instead of a mirrored surface.

Liquid crystals share with LED's the advantage of low-voltage operation. They offer the additional advantage of very low power consumption. They are easily fabricated into arrays using techniques which are closely related to those used for integrated circuit production.

LCD's show promise as display devices for graphic arts use. Arrays of liquid crystals with 30 or 40 dots is each direction are commercially available from two Japanese companies (Epson and Seiko) and much larger arrays have been demonstrated by these companies and others. The first large-volume use of LCD arrays is the hand-held "Microvision" games. These use an LCD array for games similar to the video games available for use with a TV set.

There is no doubt that LCD arrays with sufficient capacity for graphic arts use will be developed. Probably they will initially be used in consumer products such as portable TV's and games. They will be compact and light-weight and will consume far less power than CRT's. In the graphic arts, their first application will probably be portable terminals.

LCD's do have their limitations, however. One important one has to do with size. The manufacturing methods currently being used for arrays do not scale up very well, making large displays difficult to produce. Two or three inches on a side is the most likely format for the near term. Since LCD's are transparent, it has been suggested that a system analogous to a slide viewer or a rear-projection screen arrangement might be used to produce a larger image.

Another drawback to LCD's is their slow response time. To turn a dot on or off requires about a tenth of a second, which would lead to serious "smearing" during scrolling, movements of copy blocks, etc. Some promising research work has been done on ways of speeding up the LCD response, so this

with LCD's, although various schemes to achieve them have been demonstrated. For the time being, this will be a monochrome technology in most applications.

Vacuum flourescent display. Another possible contender in the flat-panel display field is the vacuum fluorescent display. This device is analogous to the CRT in several respects: the imaging process involves electrons travelling from a cathode through a vacuum and striking a phosphor. But in the case of the vacuum fluorescent (VF) display, the entire back surface of the device is the cathode. On the inside surface of the front glass is an array of independently-selectable transparent anodes. Each is coated with phosphor. When a particular anode is turned on, it attracts electrons from the cathode. These strike the phosphor, causing a bright spot.

This technology has the drawback of requiring high voltages, and the power consumption is comparable to a CRT. On the other hand, the manufacturing process is straightforward and can readily accommodate to special requirements relating to color, size, etc.

Panels based on this technology are available commercially in sizes up to 256 by 256 dots, which is more pixels than any other flat panel technology except plasma. However, the cost of these panels (over $500 each even in thousand quantities) makes them uncompetitive with CRT's in most applications. The high cost stems largely from the cost of assembling the support electronics. These electronics are not highly integrated because they handle high voltages. This technology, like the plasma panel, requires the development of high voltage integrated circuits to become competitive.

Plasma panels. The plasma panel may be thought of as thousands of tiny neon tubes sandwiched between two glass plates, one carrying vertical rows of electrodes, the other carrying horizontal ones. When the proper signal is applied to a given horizontal and a given vertical electrode, the "dot" at the intersection of the two electrodes lights. Once lit, the element can be maintained in the "on" state by a voltage far lower than that required to turn it on. This property means that given the proper driving circuits, plasma panels have a built-in "memory": dots that are lit stay lit without further attention. In this respect, the plasma panel is a little like the storage tube. However, unlike the storage tube, elements can be selectively turned off as well.

At first glance, the plasma panel would seem to be well suited to graphic arts requirements. In fact, Dymo used to offer a plasma panel typesetter preview screen, and there are still word processors, data processing terminals and computer-assisted-instruction terminals which use this technology.

Unfortunately, the plasma panel has several drawbacks. The greatest is the relatively slow response time of the individual cells. They require about a tenth of a second to change state, so updating the display is a slow process, not well suited to applications involving scrolling and moving copy around. Beyond this, there is the problem of the high voltages required by the driving electronics (too high to be handled by ordinary integrated circuits) and high power consumption. The interconnection problem, which plasma panels share with other flat panel technologies, is also a difficult one. Plasma's greatest single virtue—its inherent memory—is no longer a significant factor because of declining memory costs.

Still, research into improving plasma panels continues and answers to some of the problems—particularly, integrated circuits that cope with the necessary voltages—may be at hand. The largest firm now involved with plasma displays is Burroughs, which uses them in terminals for some applications (none in the graphic arts) and sells them to other companies. Texas Instruments recently dropped its plasma display program.

Electroluminescent (EL) displays. Certain substances give off a glow when current passes through them. LEDs do this, of course, but only in a restricted, specially-fabricated region. EL materials glow all over. For many years, a common use of electroluminescence has been in flat panel night lights.

Electroluminescent displays, like plasma panels and VF displays, require high voltages and thus can't be driven by today's standard integrated circuits. Sharp, in Japan, has developed a 240-by-320-dot EL panel which it is now marketing. It incorporates a number of novel elements, including special high-voltage integrated circuits mounted on flexible circuit "films" (instead of the usual rigid board) which are directly attached to the EL panel itself.

The power consumption of the EL panel is a great deal higher than it is for LCD's. The Sharp panel consumes about ten watts (vs. perhaps a watt for a comparable LCD, if one were available). On the other hand, Vacuum fluorescent panels are even higher in their power consumption.

The future of flat panel displays. The displays listed above are either already in commercial use or will be available for use within the next year or two. Advances in integrated circuit technology is bringing costs down and some of these technologies will soon begin competing with the CRT in certain applications. The initial applications will be ones where portability, compactness, or low power consumption are important. Cockpit displays for aircraft is one such application. Hand-held television receivers is another. Portable terminals will probably be the first graphic arts application area. It is important to note, however, that further development of the CRT is still occurring. CRT's with size and power consumption comparable to some of these competing technologies have been developed by Sinclair Research in England and are expected to reach the market (in the form of hand-held TV's) late this year or early next year.

The future of display technology

As we look at the rapidly-changing world of display technologies, we are struck by the fact that many of their limits are to be found, not in the displays themselves, but in the support electronics. The reason a display ad terminal doesn't show true fonts is not that the CRT is not up to the task. It is rather that more than an order-of-magnitude increase in font memory and character-generating processing power (plus a corresponding push in digitization efforts by the vendors) would be necessary if responsiveness were to remain at today's levels. The reason that the operator of a color page-makeup terminal frequently can't "undo" a change made several steps earlier has nothing to do with the CRT. It has to do with the limited amount of memory available for storing images. And the reason a flat panel TV is not available today is not that the display technology is unavailable. It is that the support electronics and attachment procedures have not reached commercial viability.

We occasionally hear it said that the manufacturers of integrated circuits are worried that markets will not be found for their ever more complex and powerful chips. We think they do not need to worry just yet—the field of display electronics can use everything they can offer. For the next few years at least, advances in the semiconductor state-of-the-art will be directly translatable into better, more flexible and more responsive display technology.

George A. Alexander

EKC 002002533